Kyle Xiao

March 6th, 2019

LMC 4702

Undergraduate Thesis

Machine Learning based Procedural Content Generation in Semantic Choreography

Abstract

BeatMania is a rhythm-action game where players press buttons in response to *keysound* events to generate music. Rhythm-action game charts (the sequence of keysound events) have traditionally been human authored, since each song level must be creatively organized and correspond an overall pattern or theme. A deep neural network approach is proposed for rhythm-action game chart creation, and a method of level evaluation for co-creative AI is defined. That is, given an arbitrary piece of music, human users can generate BeatMania charts as well as give input to an AI collaborator. The problem is divided into two parts: autonomous chart generation and design interaction. For the chart generation process, a combination of features that include grouping information and audio sample labels are incorporated into an artificial neural network. For the design interaction, principal component analysis is utilized for a proposed reinforcement learning model. The co-creative tool is tested against Markov Chain and LSTM baselines via human trials.

INTRODUCTION

Procedural content generation via machine learning (PCGML) is an area of research that focuses on the generation of game content, such as level design or other creative artifacts, using algorithmic means. The field has become more prominent, especially in game development and technical machine learning research. PCGML by itself is employed to enhance game replayability and reduce production costs (Summerville et al 2017), but also in a broader sense is a domain in which some of the cutting edge in machine learning can indefinitely enable uniquely "creative" gameplay experiences. PCGML techniques have been rapidly evolving due to novel machine learning techniques and open-sourced datasets, especially with the resurgence of *deep learning* as a method to build models from big data. Some of the cutting edge in machine learning include applying *general adversarial networks* (GANs), n-grams, Markov models, autoencoders, and reinforcement learning to enhance traditional algorithms, though in PCGML they have each been used only in narrow contexts for certain games as techniques are often domain specific.

Co-creativity, the collaboration of human and AI in creative endeavors, is an open problem space in PCGML explored through mediums such as art, humor, games, and robotics (Karimi et al. 2018). In contrast to autonomous PCGML, the primary focus of co-creativity has been to allow computers and human users to interact with each other and contribute to creative artifacts. This enables objectives like encouraging authors with little experience to contribute, facilitating the process of creation and review, and making products with a cohesive theme that fits the author's vision. Examples include Drawing Apprentice for visual art (Davis et al. 2015), ViewPoints AI for performing arts (Jacob and Magerko 2015), and Sentient Sketchbook for level design (Yannakakis, Liapis, and Alexopoulos 2014). Some of the limitations of co-creative systems, though, is related to how to evaluate feedback on creative work. Previous approaches in co-creativity explore methods such as constraint-satisfaction and evolutionary algorithms by providing examples in the target domain (Summerville et al. 2017), but directly utilizing semantic human feedback to inform AI decisions has yet to be explored. For instance, utilizing qualitative feedback as a reward signal in *reinforcement learning* is a promising area in the do-creative domain.

The aim of this thesis is to explore PCGML techniques in the domain of chart generation for *Beatmania IIDX* (BMIIDX), a rhythm-action game where player hit keys in response to cues to make music, as well as to propose a dataset and method of evaluation which sets a basis for co-creative reinforcement learning. A *chart* is a set of actions known as *keysounds* timed to music that in other games can correspond to dance moves or hitting an instrument key. The process of chart choreography for rhythm-action games has historically been hand-crafted, as human authors must make creative and logical decisions as to how to place events such that the chart accompanies the music while also balancing difficulty and technical themes. Previous work in chart generation assumes an *a priori* unknown piece of music in domains like *Dance Dance Revolution* (Donahue et al 2017), whereas this paper addresses informing a PCGML model given an existing musical score. This implies that although there has been previous work in rhythm-action PCGML, the specific challenges addressed in this research are novel. Afterwards, a set of evaluation metrics is standardized to inform a reinforcement learner, based on a semi-Markov Decision Problem (Sutton et al. 1999).

In the first part of the paper, autonomous PCGML in chart generation is addressed. In BMIIDX, a *playable* note is defined as specific timestamped note which the player is expected to perform an action to play, whereas a *non-playable* note is played automatically in the background. In addition, both playable and non-playable notes have a one-to-one correspondence with an audio sample. Thus, a classifier model is proposed to categorize playable and non-playable notes to generate a chart. The proposed model is a feedforward artificial neural network, which takes as input for each note the relative measure information, grouping with other classified notes, instrument classification, and pitch information. The instrument classification is generated via a convolutional neural network, which analyzes the audio sample spectrogram to label sounds as one of 10 instrument classes. In addition, an LSTM and naïve baselines are provided for comparison.

Following autonomous chart generation and ground truth-based assessments, human subject feedback is used for evaluation. The purpose of this evaluation is to qualitatively asses the autonomous model performance as well as to gather a dataset for a co-creative model, where a human feedback loop in chart generation can be added for future work. In this study, the data is collected via a level design editor and a survey. A set of qualitative characteristics for evaluating a chart is defined (e.g. is the chart interesting or natural), and these characteristics are compared against several other baselines. Subjects are asked to rank both generated and hand-authored charts based on the set of characteristics in the hypothesis space and are asked to make improvements to the charts as they see fit. Finally, a statistical analysis is run to find consistencies and correlation among responses. Using the gathered data for axes with high correlation, principal component analysis is utilized to narrow dimensionality into a signal that can be used as a reinforcement-learning reward. This process is compared across multiple models and chart authors.

Literature Review

Methods for procedural content generation (PCG), the creation of game content via algorithmic means, have become increasingly prominent in both commercial game development and in experimental research for human-virtual interactions (Summerville et al 2018). Academic PCG research addresses how PCG can enable new player experiences, including generating material which can more deeply interact and adapt to the player. Because of advances in AI and the recent availability of large datasets through open-source projects, generative models have become more powerful, allowing for the advent of PCGML. Among research in PCGML, one of the most compelling use cases is in co-creative design, where users interact with an AI agent to contribute toward a creative artifact (Karimi et al 2018). Though research in the fields of PCG and ML respectively have already had considerable forward progress, the intersection of the two has the potential to yield promising results for co-creative systems.

Past Approaches to Co-Creative PCGML

Past approaches to co-creative PCGML employ methods like constraint satisfaction, a search algorithm that finds a set of parameters conforming to domain-specific or user-imposed rules, and evolutionary algorithms, a set of techniques that stochastically generate and update models based in a fitness function. One such project is Sentient Sketchbook, a computer-assisted game level authoring tool, which provides an approach utilizing genetic constraints (Liapis et al 2014). Sentient Sketchbook uses a mixed-initiative approach, which implies that the AI agent proactively initiates suggestions to the human author as he or she is editing the level. This is inherently co-creative, since the agent actively contributes to the creative artifact. The algorithm employs a genetic approach, which increases probabilistic weights in the sampling distribution of models which score higher on a fitness function. However, it also utilizes novelty search under constraints, which places emphasis on adding diverse parameters to the distribution that meet a predefined minimal criterion.

Another approach is the Super Mario Bros co-creative level design editor (Guzdial et al 2018). In this study, Markov Chains, Bayes Nets, and LSTMs are compared as models which infer future changes a human user might make in the context of a Mario level-editor. The Markov chain attempts to model the user interaction as a series of probabilistic states, giving more weight to local coherence than LSTMs, which form a more global chain of dependencies. The Bayes Net, in comparison, acted as a baseline for the other approaches, as it was the focus of a previous study by the same researcher. In Sutskever et al 2014, it is shown that local coherence via Markov Chains tended to yield better qualitative results in the user study, suggesting that for generalizing to other co-creative applications, models that can learn information based on the "closeness" of features may be more effective.

Relevant ML Literature

Though literature directly related to co-creativity has seen some exciting work, novel approaches in other fields of ML have potential applications in PCGML. Most prominently, artificial neural networks (ANNs) have been wildly successful in most ML fields with extensive data, including generative tasks. For instance, deep computer vision research on perceptual style-transfer propose interesting techniques that could be used to build more subjectively diverse models (Alahi et al 2016). These models utilize a convolutional neural net (CNN) to down-sample and then up-sample windows on an image and then form a generative model trained with feature reconstruction loss determined by a separate CNN.

Other approaches utilize GANs, which could directly be used as an engine for co-creation. GAN architecture consists of a generator and discriminator, where the generator tries to increase error in the discriminator while the discriminator tries to distinguish between real samples and those made by the generator. Some of the most recent developments in GAN technology include the use of custom architectures, such as Masked Residual Unit and Gated Recurrent Units (Chen and Hays 2018). Though most research focuses on image generation, similar models can be mapped to discrete tile mappings or other game state representations.

Certain game levels can be represented as encoded sequences, making recent work in sequence-to-sequence generation relevant to PCGML. Work in machine translation models, for instance, aims to improve LSTMs for language processing (Sutskever et al 2014). Like the findings of Guzdial et al, minimizing long term dependencies and increasing local coherence greatly improves model performance, as shown when comparing performance in reverse word order sentences. Unlike Guzdial et al, though, LSTMs outperformed their baseline (SMT), which is likely because of the difference in problem space.

Intersection of Co-Creative PCG and Novel ML Techniques

In the current state of PCG and ML, some of the newest ML techniques have yet to be tested for co-creative applications. The bulk of existing co-creative software is written on techniques that do not take advantage of the large amounts of recently available data (Summerville et al 2018). These algorithms include classical search, constraint satisfaction, and genetic algorithms that have already reached a bottleneck in terms of performance. The wave of new research in ANNs, on the other hand, has been shown to produce groundbreaking results in image generation, machine translation, and style transfer.

Although these ML techniques have been shown to yield impressive results in their respective fields, there has yet to be work that directly applies them to co-creative AI. Of the very recently published literature that does examine certain deep learning techniques in co-creativity, the scope has been relatively limited. Certain elements, such as learning semantic representation of style or learning patterns from sequence encoded game states, have yet to be thoroughly explored. In addition, other process engineering and model architectures which exploit novel features have yet to be tested for PCG. These are challenges which have yet to be addressed by co-creative PCGML research, though this thesis aims to explore some of these problems in the rhythm-action game domain.

Methodology

There are several components to the chart generation process. The first is the acquisition of cleaned and labeled training data as well as the data aggregation into a centralized database. Secondly, there are the supporting models that generate features for an input chart. Third, there is the feedforward neural network model that combines semantic features and outputs the playable/non-playable classification. Lastly, there is the process of assigning controls for the set of playable notes. In addition to the chart generation, there is the methodology for conducting human trials of chart review, which is designed so participants compare, score, and rank charts from human-authored or generated sources. Finally, a statistical analysis is performed on response correlation and principal component analysis and propose a model for a reinforcement learning agent.

Data Generation

Raw chart data is acquired from *BMS of Fighters* (BOFU), an annual community event that compiles a list of novel hand-authored BMIIDX charts. The BOFU 2011 dataset, which we refer to as “BOF2011”, was utilized to generate the dataset and consists of 1,454 charts for 366 songs. There are 171,808 unique audio samples, 1,242,394 playable objects, 4,320,683 total objects, and an average of 3.97 charts per song. Data is captured for each chart, including playable classification, note placement, and sound binary files, into a JSON format parsed in a MongoDB database.

A second dataset, BOFU 2015 consisting of 60,714 samples, is used as training data for an audio classifier, which categorizes audio samples into a set of 10 instrument labels. For example, whether a certain sound file is a piano, drum, or synth sound. However, there are no consistent labels for this dataset as authors have free discretion to arbitrarily name their audio samples, so inferencing was made based on matching the filenames to a set of common labels which were determined by the most commonly occurring substring labels out of a dictionary of instruments. This data was then fed into the audio sample classifier, which gives labels as a supporting feature.

Feature Generation

A set of semantic features is identified, which are domain-specific metrics that are combined as input to the model to make the final inference. The first feature generation model consists of *sample classification*, which is the process by which audio samples are identified as belonging to certain instrument classes. A chart consists of sample-time pairs that point to raw audio files and learning semantic information about the audio files provides key data from recurring instrumental archetypes. To form the model, the data is preprocessed into a spectrogram representation and transform it into “audio fingerprints,” which is a vector of mel-frequency cepstral coefficients (MFCC) (Logan et al 2000) of the log-magnitude mel-scale wave amplitudes over time. The bit rate of the sound is also fixed to 16k, so that the representation has a consistent temporal resolution. The fingerprints are then fed through a two-layer 2D convolutional neural network. Each layer is activated via a Rectified Linear Unit (ReLU) and then max-pooled. A gradient descent optimizer is utilized with a 50% dropout rate, a step size of 0.01, and cross entropy loss. In addition, pitch information is also encoded as a separate feature via the PySound library.

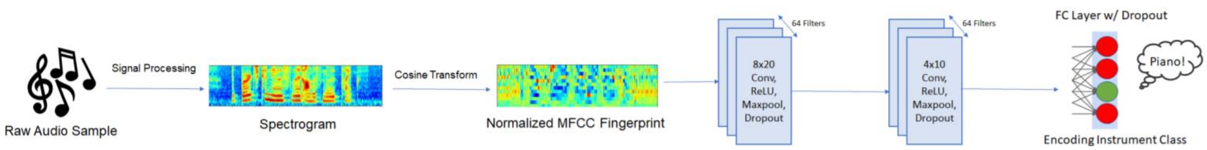

Figure 1: The model for classifying sound files into corresponding instruments

The second feature generation model is *challenge modeling*, which is the process of mapping chart difficulty over time. Note placement for different difficulties vary widely, as more difficult charts are more playable note dense. To compute difficulties within the training set, a rule-based technique is used as defined from the *Osu!* rhythm action game. The difficulty for each given chart at any moment in time is the weighted sum of *individual strain* and *overall strain*. Individual strain is the interval between playable notes mapped to the same control on an exponentially decaying scale. Overall strain is defined as the number of controls which must be activated simultaneously. The overall difficulty is generated via the weighted sum of the highest local strain values throughout the chart.

Finally, other domain-specific features are derived. These features include *measure information*, *summary*, and *grouping*. Measure information is the beat alignment, which is a value from 0 to 15 representing where in a measure or bar a note is placed. For instance, for a chart with 4/4 time signature and 150 BPM (beats per minute), one unit represents around 25 milliseconds. Summary is a 1x270 vector the encodes the playability of samples before the current object. Five time windows were used, covering 2, 4, 8, 16, and 32 beats, each of which has a 1x54 time window that gives probabilities for playable/nonplayable classifications based on observed appearances.

Feedforward Model and Sample Selection

For the combined feedforward network, the features are placed as input to predict playable and non-playable classifications. For the neural network modules, a mean squared error (MSE) with weight of 1 to playables and 0.2 to non-playables is utilized to learn parameters. The model is trained on an Intel i7-5820K CPU and NVIDIA GeForce 1080 GPU. In addition to our combined feedforward network, an LSTM baseline is trained as inspired by Donahue et al 2017, though changes were made in order to translate the model into a keysound-based game like BMIIDX.

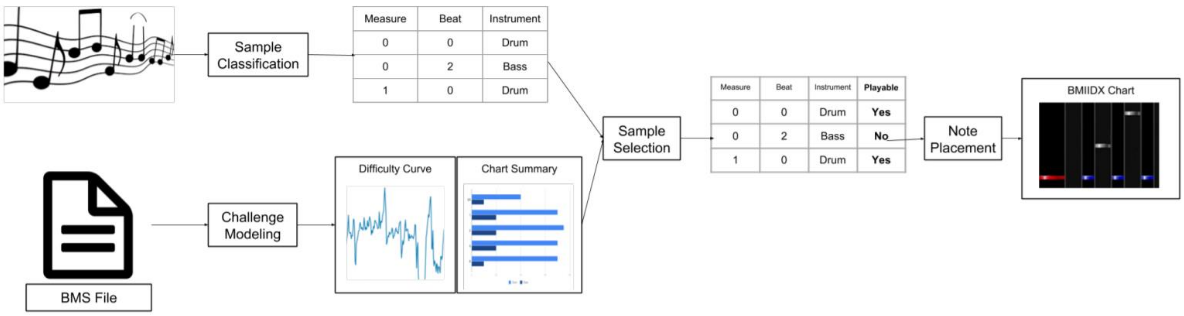

Figure 2: Combining the generated features into the final model

Note Placement

Once the playable/non-playable classification is made, the playable notes must be mapped to one of 8 controls to generate a coherent chart. At minimum, such a mapping need only to ensure no two playable notes at the same timestamp are mapped to the same control. A simple module is defined that replicates our feedforward models and features for sample selection, but instead is trained on control labels. In addition, a post-processing step is added that double checks the constraint that no simultaneous notes are mapped to the same control in too short of a time interval. If there exists such a violation, an arbitrary free control is selected instead, though this is a rare occurrence.

Human Study Methodology

The autonomous chart generation model is tested against human subjects. A survey is conducted with the described model generated charts, human authored charts, randomly generated charts, and the LSTM baseline model. Participants are asked to rank and score the charts on several semantic attributes, which are:

- How interesting

- How natural
- Consistency in challenge
- Ability to convey musical ideas
- Replayability

Participants are then asked what changes they would make to a chart, as well as what aspects they found enjoyable. Scores are recorded out of 10 as well as recommended changes.

Statistical Analysis

Utilizing the gathered data, dimensionality reduction is performed among each of the queried attributes. Principle component analysis (PCA) is utilized, which is a method for performing a linear mapping of data into a lower dimensional space. The covariance matrix is examined to determine the fewest dimensions which capture the most complexity and choose an n-dimensional representation that captures over 90% of variance. A semantic meaning is assigned based on observed interpretations as well as qualitative responses to the survey, and the resulting numerical values represent a mapping which can be used in evaluation for a reinforcement learner reward signal.

Results

Displayed below are the results for playable classification experiments, presented in mean and standard deviation. FF, BP, CM and RS refer to Feedforward, Beat Phase, Challenge Model, and Relational Summary, respectively.

Model	F1-Score	Precision	Recall
Random	0.291±0.089	0.335±0.200	0.299±0.020
All Playable	0.472±0.207	0.335±0.199	1.000±0.000
LSTM + Audio Features + BP + CM	0.424±0.154	0.767±0.176	0.353±0.248
LSTM + Audio Features + BP + CM + RS	0.499±0.225	**0.805±0.121**	0.405±0.237
FF + Audio Features + BP + CM	0.253±0.143	0.523±0.266	0.179±0.113
FF + Audio Features + BP + CM (Self Summary)	0.368±0.198	0.422±0.213	0.392±0.258
FF + Audio Features + BP + RS	0.621±0.206	0.760±0.110	0.568±0.254
FF + Audio Features + BP + CM + RS	**0.698±0.162**	0.778±0.112	**0.649±0.197**

Table 1: Results of model and baselines via F1 scores

Model performance is measured via *F1 score*, which is the harmonic mean of *precision* and *recall* on ground truth labels in the dataset. Precision and recall are defined as the ratios of true positives to retrieved elements and true positives to relevant elements respectively.

Across all models, relational summary data improved F1 score performance, and in the feedforward network challenge modeling improved performance by 7.7%. Interestingly, the feedforward network outperformed against all LSTM baselines in terms of F1 scores. This is noteworthy since an LSTM is a recurrent model, meaning temporal information should be embedded in the model by default. However, the complete LSTM model still outperformed in terms of precision with statistical significance at $p < 0.025$ (N= 145), though both precision and recall are important for chart generation.

Full Feedforward Model Performance on Charts

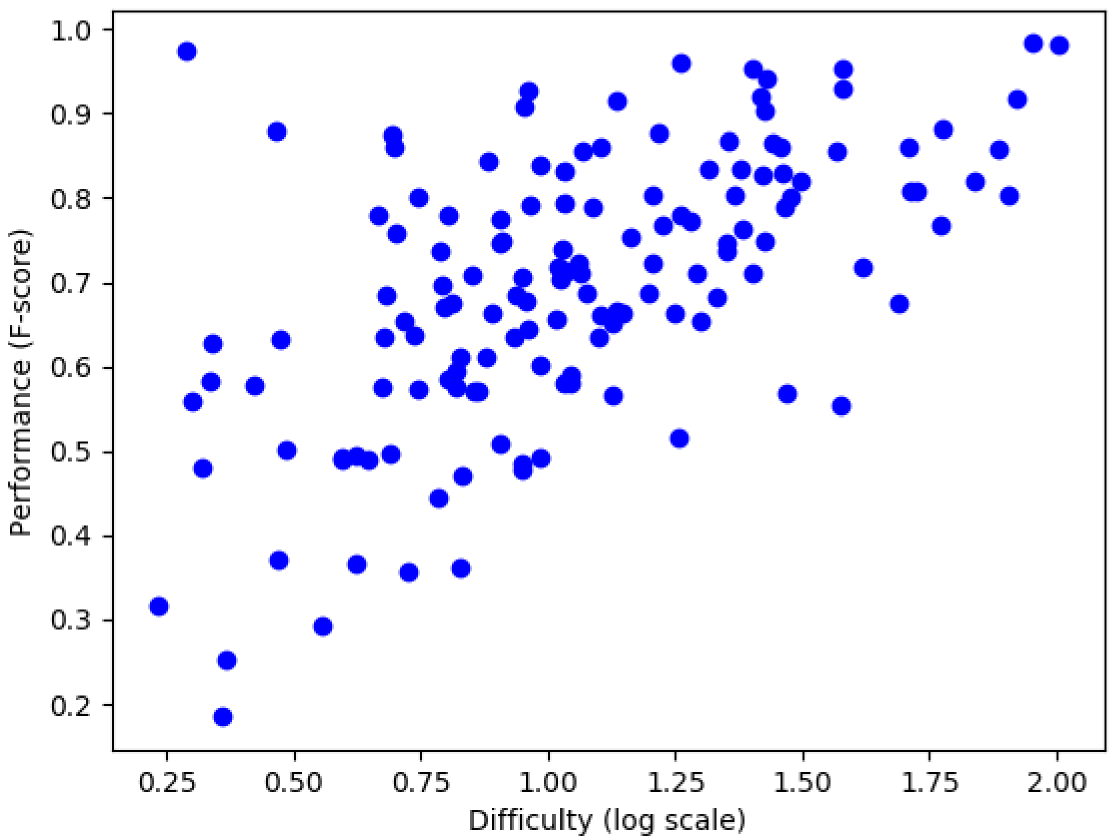

Figure 3: Results of model over chart difficulty

Like the findings of Donahue et al 2017, performance generally decreases with less difficult charts. The full model achieves consistently high F1 scores for high difficulty but has high variability on more simple charts. In addition, testing with other features such as sound event density, audio pitch, number of instrument classes, and hierarchal measure representations failed to improve performance and, in some cases, caused the model to perform worse.

***Reader's Note: The human subject study data is still pending, but will be inserted here*

Discussion

Throughout the described experiments, there were several key observations. In particular, the feedforward model with the semantic features outperformed LSTM baselines. This is significant since this implies our relational summary feature was able to better capture temporal dependencies than sequence-based recurrent modeling. Though the LSTM can directly observe history information of playable and nonplayable labels that the model has inferred, the relational summary only imparts history information using summary statistics. Thus, the feedforward model likely performed better due to the high variance in charts, thus causing the LSTM to perform worse due to the degree of noise.

However, though the feedforward network had a higher F1 score, the LSTM model still achieved a higher precision with statistical significance, though at the expense of recall. Intuitively, this means that the LSTM is pickier with playable notes but misses out on many notes that should be playable. Though it is unclear why this is the case, a possible explanation is that exposing the classification to more granular data such as in a recurrent neural net gives higher confidence for examples where features strongly agree, but also opens cases where there are conflicting signals or where the model has difficulty generalizing to a novel combination of features. This is especially a problem for high variance data, as mentioned previously.

Another observation with the feedforward network with challenge modeling is that lower difficulty charts have high variance in F1 scores. Donahue et al 2017, which showed similar findings, posits that “this owes to the comparative class imbalance of the lower difficulty charts.” We also hypothesize that there is more variance in how authors choose playable notes in less difficult charts, since the sparsity of playable events allows more freedom in choosing playable notes.

For the note placement evaluation, the F1 score metric as used in playable classification evaluation is not a representative measure for this context. This is because no baseline exists for placing non-playable notes. In addition, since evaluation is per-note based, a naïve evaluation metric would have difficulty capturing patterns where groups of notes are placed in a certain order or shape.

***Reader's Note: The human subject study discussion will be inserted here. In addition, this last paragraph may change depending on the results of the human subject study if any new qualitative ways to measure note patterns becomes apparent.*

Conclusion

Rhythm-action game choreography is a difficult problem, and by using semantic features this paper has demonstrated a pipeline that can outperform existing models. The work done in this experiment was on Beatmania, a keysound-based game which has notes that can either be playable or non-playable. This research also established a novel chart dataset by aggregating raw data from the BOFU collections. The autonomous generation model allows for control over difficulty progression and stylistic choice depending on challenge modeling and feature selection respectively.

The experiments demonstrate that a feedforward network with challenge modeling and relational summary can outperform traditional temporal-based models such as LSTMs in terms of playable and non-playable classifications (i.e. whether a note in a chart should be a player-triggered event or played automatically). The model is also open to stylistic blending, as both challenge models and relational summaries from different charts can be swapped in the generation to create novel charts.

By utilizing some of the cutting edge in PCGML techniques, we show state of the art results in game chart generation. The model can be of great benefit to the homebrew chart authoring community, as well as to novice game designers. Both the data and described methods are open sourced and could hopefully help spur future innovation related to game content generation.

***Reader's Note: Again, perhaps another page will be added once the human subject results are in*

Future Work

The work done in this paper highlights the current attempt at autonomous chart generation via PCGML with an emphasis on deep learning and semantic feature generation. The immediate future work to be done would involve improving the existing model. For instance, in Donahue et al. 2017 delta-beat information was incorporated, which is a feature that measures the number of beats between steps. However, this feature was not included in this model since classification is done on individual playable/nonplayable notes so unrelated notes can be placed on short or simultaneous time periods. This information could be encoded via a grouping method for certain strings of notes, which has yet to be experimented with.

In addition, the current challenge model is relatively simplistic and could be expanded upon. The current model is a rule-based system which assumes perceived difficulty is the same across all players for a given string of notes. In addition, due to being a rule-based algorithm the model is highly sensitive to parameter tuning, meaning further optimization is possible. A model-free approach via a hidden policy or a player experience-based system could be solutions, but these approaches have yet to be tested.

Finally, having the basis for an experience-based reinforcement learning framework, the next step after autonomous chart generation is introducing computational co-creativity in practice. The current feed-forward model allows generation on the fly, since classifications are made on a per note basis. This allows for a dynamic chart generation process where user input can be encoded as a feature. In addition, deep-q networks (DQNs) can be introduced, whereby our feedforward network is instead trained on q values with human ratings as the reward functions. The same semantic features can be utilized, but the creative expression would be a conjunction of human and artificial inputs.

Glossary

- ANN (Artificial Neural Network) – A hierarchal machine learning model that approximates a function by using several linear and nonlinear composition layers in sequence. Originally inspired by biological neural networks.
- Autoencoder – An ANN model where the output ground truth is the same as the input. This is done to encode data efficiently by reducing the dimensionality of middle layers, thereby representing complex data with fewer parameters.
- Bayes Net – A probabilistic directed acyclic graph model that encodes conditional dependencies. Often used to predict and update likelihood of variables based on observations of the environment.
- BMIIDX (BeatMania IIDX) – A rhythm action game where players respond to keysounds.
- BMS – The file encoding of a BeatMania chart.
- BOFU (BMS of Fighters) – An annual community event where authors compile BMMIIDX charts.
- Chart – A sequence of keysound events where players respond of playable audio cues. This is analogous to a stage or level in other games.
- CNN (Convolutional Neural Net) – An ANN that incorporates sliding kernels as a feature, often used in image processing. A kernel in this context is a window or crop in an image with a mask determined by a set of learned parameters.
- Co-Creativity – The process by which a human and an AI collaborate on a creative artifact.

- Constraint Satisfaction – A search algorithm which finds a set of valid parameters given a set of domain-specific or user-imposed rules.
- Cross-Entropy – A loss metric used in training models, which measures how badly a model is doing for a classification task.
- Deep Learning – The class of ML models that utilize hierarchal learning. The most prominent deep learning models are ANNs.
- Dropout – A form of regularization during ANN training where nodes are zeroed out. This is done so the ANN doesn’t depend too heavily on any small subset of nodes and overfit the data.
- Feature – An input that a model uses to make a prediction.
- Feedforward Network – A type of ANN which transfers data between layers in a linear order.
- GAN (Generative Adversarial Model) – A set of two models where one acts as a generator and one acts as a discriminator. The two models work against each other in an arms race fashion, and often the generator is taken to create synthetic data.
- Genetic Algorithm – An algorithm which generates and evaluates models based on a fitness function, and then increases the sampling distribution for parameters that score well.
- Gradient Descent – An optimization algorithm which attempts to minimize loss by using first-order Taylor approximations of the derivative.
- Ground Truth – The correct or underlying label or mapping for a set of data.
- Individual Strain – A metric which is defined as the interval between playable notes on the same control, which is weighted more heavily for more recent consecutive sequences.

- Keysound – An action which consists of a timestamp and sound in which a player may respond to.
- LSTM (Long-Short Term Memory) – A type of RNN which utilizes memory cells. Often used as a baseline technique in sequence to sequence generation.
- Markov Decision Problem – A time discrete framework for mapping certain classes of problems. Primarily used in decision making problems, it assumes discrete state spaces with transitions that are partially random.
- Markov Chain – A stochastic model that assumes the probability of an event depends only on the previous event.
- Measure Information – A set of metrics that represent where a note falls in a discrete subdivision of musical time.
- MFCC (Mel-frequency Cepstral Coefficient) – A representation of the short-term power spectrum of a sound. A sound is mapped to a cosine transform, or a linear combination of log power spectrum, and the coefficients make up the MFCC.
- ML (Machine Learning) – The study of algorithms that perform tasks without explicit instructions, instead relying on data-driven patterns.
- Model – A system which encodes a set of assumptions about a process which can then be used to query data or make inferences.
- MSE (Mean-Squared Error) - A loss metric used in training models, which measures how distant a model output is from the ground truth.
- N-gram – A model that uses a continuous sequence of n items from a larger sequence, which is then used to make probabilistic inferences for following elements.

- Overall Strain – A metric for difficulty defined as the number of controls that must be activated at the same time.
- PCA (Principle Component Analysis) – A statistical analysis technique which reduces high dimensional data into a smaller set of constructed dimensions that encode most of the original data.
- PCG (Procedural Content Generation) – The creation of game content without explicit instruction from a human author.
- PCGML – The study of applying ML techniques to accomplish PCG.
- Playable/Non-playable – A classification for whether a keysound should be explicitly be made into an event that a human player must act upon. The objective of the player is to hit all controls for playable keysounds at the correct timestamps to play music, whereas nonplayable notes are played automatically in the background.
- Reinforcement Learning – A model which learns a policy based on reward signals. It concerns how agents act in an environment to maximize cumulative reward.
- ReLU – A nonlinear activation function which is defined as ReLU(x) = max(0, x)
- Rhythm-action game – A type of game where players hit keys in response to cues or events with some music.
- RNN (Recurrent Neural Net) – A type of ANN where node connections are made as a directed graph in a temporal sequence. Internal states are reused as inputs for the next step when processing a sequence of inputs.
- Semantic Feature – A feature which is specific to the domain and is human interpretable.
- Summary – A feature that aggregates semantic features of past states in a sequence.

Sources

Alemi, O.; Francoise, J.; and Pasquier, P. 2017. GrooveNet: Real-Time Music-Driven Dance Movement Generation using Artificial Neural Networks. networks 8(17):26. Chan, A. 2004. CPR for the Arcade Culture.

Chen, W., and Hays, J. 2018. SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis. CoRR abs/1801.02753

Donahue, C.; Lipton, Z. C.; and McAuley, J. 2017. Dance Dance Convolution. In Proceedings of the 34th International Conference on Machine Learning.

Guzdial, M., and Riedl, M. 2016. Game level generation from gameplay videos. In Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference.
Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. Neural computation 9(8):1735–1780.

Hoover, A.K.; Togelius, J.; and Yannakis, G. N. 2015. Composing video game levels with music metaphors through functional scaffolding. In First Computational Creativity and Games Workshop. ACC.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision, 694–711. Springer.

Karimi, P.; Grace, K.; Maher, M. L.; and Davis, N. 2018. Evaluating Creativity in Computational Co-Creative Systems. ArXiv e-prints.

Machado, T.; Gopstein, D.; Nealen, A.; and Togelius, J.2018. Ai-assisted game debugging with cicero.

MILLER, K. 2009. Schizophonic Performance: Guitar Hero, Rock Band, and Virtual Virtuosity. Journal of the Society for American Music 3(4):395429.

Mogren, O. 2016. C-RNN-GAN: continuous recurrent neural networks with adversarial training. CoRR abs/1611.09904.

Nogaj, A. F. 2005. A genetic algorithm for determining optimal step patterns in Dance Dance Revolution.

Sainath, T. N., and Parada, C. 2015. Convolutional neural networks for small-footprint keyword spotting. In Sixteenth Annual Conference of the International Speech Communication Association.

Smith, G.; Treanor, M.; Whitehead, J.; and Mateas, M. 2009. Rhythm-based Level Generation for 2D Platformers. In Proceedings of the 4th International Conference on Foundations of Digital Games, FDG’09, 175–182. New York, NY, USA: ACM.

Summerville, A.; Snodgrass, S.; Guzdial, M.; Holmgard, C.; Hoover, A. K.; Isaksen, A.; Nealen, A.; and Togelius, J. 2017. Procedural Content Generation via Machine Learning {(PCGML)}. CoRR abs/1702.0.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., Advances in Neural Information Processing Systems 27. Curran Associates, Inc. 3104–3112.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. Artif. Intell. 112(1-2):181–211.

Zhu, J.; Liapis, A.; Risi, S.; Bidarra, R.; and Youngblood, G. M. 2018. Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation.

Zook, A., and Riedl, M. O. 2015. Temporal game challenge tailoring. IEEE Transactions on Computational Intelligence and AI in Games 7(4):336–346.